

## □ JavaScript Loops & Methods – □□□□ Notes (□□□□□ □□□)

### □ 1. for loop

□ □□□□□□□□: □□ index □□□□□ □□ □□ manually control □□□□ □□

□ Return: □ □□□ return □□□□ □□□□

□ Structure:

```
for (let i = 0; i < array.length; i++) {  
  // □□□  
}
```

### □ 2. while loop

□ □□□□□□□□: □□ □□□ □ □□ □□□□□ □□□ □□□□□ □□

□ Return: □ □□□ return □□□□ □□□□

□ Structure:

```
while (condition) {  
  // □□□  
}
```

### □ 3. do...while loop

□ □□□□□□□□: □□ □□ □□ □□ □□ □□□□□ □□

□ Return: □ □□□ return □□□□ □□□□

□ Structure:

```
do {  
  // □□□  
} while (condition);
```

### □ 4. for...of loop

□ □□□□□□□□: Iterable □□□□□□ (□□□□□ Array, String) □□ values □□

□ Return: □ □□□ return □□□□ □□□□

□ Structure:

```
for (let value of array) {  
  // value □□ □□□  
}
```

### □ 5. for...in loop

□ □□□□□□□□: Objects □□ keys □□

□ Return: □ □□□ return □□□□ □□□□

□ Structure:

```
for (let key in object) {  
  console.log(object[key]);  
}
```

### □ 6. forEach() method

□ □□□□□□□□: □□□□□□ Arrays □□

□ Return: □ □□□ return □□□□ □□□□

□ Structure:

```
array.forEach((value, index) => {  
  // value/index □□ □□□  
});
```

### □ 7. map() method

□ □□□□□□□□: □□ □□ element □□ □□□ □□□□□ □□□□ □□□□□

□ Return: □ □□□ array

□ Structure:

```
let result = array.map((value, index) => {  
  return value * 2;  
});
```

### □ 8. filter() method

□ **filter()**: **filter** condition **filter** filtered array **filter**  
 □ Return: □ **filter** array  
 □ Structure:  
 let filtered = array.filter((value) => value > 2);

□ 9. **reduce()** method  
 □ **reduce()**: **reduce** array **reduce** single value **reduce** **reduce** **reduce**  
 □ Return: □ single value (number, string, object etc.)  
 □ Structure:  
 let total = array.reduce((acc, value) => acc + value, 0);

□ 10. **find()** method  
 □ **find()**: **find** matching element **find**  
 □ Return: □ **find** match □ undefined  
 □ Structure:  
 let result = array.find((val) => val > 3);

□ 11. **some()** / **every()** method  
 □ **some()** / **every()**:  
 - **some()**: □ □ □ □ element condition pass □ □  
 - **every()**: □ □ □ □ elements condition pass □ □ □ □  
 □ Return: □ true □ false  
 □ Structure:  
 let hasValue = array.some(val => val > 5);  
 let allGood = array.every(val => val < 10);

□ Quick Summary Table:

Method/Loop	Array □ □	Object □ □	Return □ □ □ □ □ □ ?	□ □ □ □ □ □ □ □ □ □ □ □ ?	Syntax Simple?	Remarks
for	□	□	□	□	□	Index control
while	□	□	□	□	□	Condition-based
do...while	□	□	□	□	□	Runs at least once
for...of	□	□	□	□	□	For values
for...in	□	□	□	□	□	For keys
forEach()	□	□	□	□	□	Simple, can't break
map()	□	□	□	□	□	Return new array
filter()	□	□	□	□	□	Useful for conditions
reduce()	□	□	□	□	Medium	Reduces to one
find()	□	□	□	□	□	First match
some()	□	□	□	□	□	At least one
every()	□	□	□	□	□	All match